Docket No.: 00-8013 RCE 1

BEST AVAILABLE COPY

COMPLETE LISTING OF CLAIMS
IN ASCENDING ORDER WITH STATUS INDICATOR

THIS LISTING OF CLAIMS WILL REPLACE ALL PRIOR VERSIONS AND LISTINGS OF CLAIMS IN THE APPLICATION.

1.   (Currently Amended) A method for generating a schema for a relational database corresponding to a document having a document-type definition and data complying with the document-type definition, the document-type definition having content particles representative of the structure of the document data, as well as loading the data into the relational database in a manner consistent with the relational schema, the method comprising the steps of:

extracting metadata representative of the document-type definition from the document-type definition;

storing the metadata in at least three tables comprising a metadata item table containing metadata representative of element types in the document-type definition, a metadata attribute table containing metadata representative of attributes in the document type definition, and a metadata nesting table containing metadata representative of nesting relationships between particles in the document type definition;

automatically generating the schema for the relational database from the metadata, wherein at least one table is thereby defined in the relational database corresponding to at least one content particle of the document-type definition via the metadata; and

loading the document data into the at least one table of the relational database according to the relational schema in a manner driven by the metadata.

2.   (Cancelled)

3.   (Currently Amended) The method of claim 1 2 wherein the extracting step further comprises the step of creating at least one default item in the item metadata table.

4.   (Previously Presented) The method of claim 3 wherein the extracting step further comprises the step of updating the item metadata table with each of the element type content particles of the document-type definition.

2

Docket No.: 00-8013 RCE 1

5.   (Cancelled)

6.   (Currently Amended) The method of claim 4 5 wherein the extracting step further comprises the step of creating a default attribute value in the attribute metadata table corresponding to attributes of element types in the document-type definition.

7.   (Previously Presented) The method of claim 6 wherein the extracting step further comprises the step of updating the attribute metadata table with each of the attribute type content particles of each element type of the document-type definition.

8.   (Cancelled)

9.   (Currently Amended) The method of claim 7 8 wherein the extracting step further comprises the step of generating a row in the nesting metadata table corresponding to each relationship between items identified in the item metadata table.

10.   (Previously Presented) The method of claim 9 wherein the generated nesting table row indicates the cardinality between a pair of items.

11.   (Original)   The method of claim 10 wherein the cardinality is one of one-to-one and one-to-many.

12.   (Currently Amended) The method of claim 7 8 wherein the generated nesting table row indicates a relationship between a parent item and a child item.

13.   (Currently Amended) The method of claim 7 8 wherein the generated nesting table row indicates a relative position of a child item with respect to other items in a definition of the corresponding parent item.

14.   (Previously Presented) The method of claim 7 wherein the generating step further comprises the step of creating at least one table in the schema of the relational database corresponding to at least one row of the metadata item table.

3

15.   (Previously Presented) The method of claim 14 wherein the generating step further comprises generating at least one default field in the table of the schema.

16.   (Previously Presented) The method of claim 15 wherein the generating step further comprises the step of altering the schema of the relational database to add at least one column to the at least one table in the relational database schema corresponding to each row of the metadata attribute table.

17.   (Previously Presented) The method of claim 16 wherein the generating step further comprises the step of altering the tables in the schema of the relational database to add columns representing links between tables of the relational database schema-corresponding to each relationship identified in each row of the metadata nesting table.

18.   (Previously Presented) The method of claim 17 wherein the generating step further comprises the step of altering the tables in the schema of the relational database by adding a foreign key to a parent table if the identified relationship is a one-to-one relationship.

19.   (Previously Presented) The method of claim 18 wherein the generating step further comprises the step of altering the tables in the schema of the relational database by adding a foreign key to a child table if the identified relationship is a one-to-many relationship.

20.   (Previously Presented) The method of claim 19 and further comprising the step of initializing a link table.

21.   (Previously Presented) The method of claim 19 and further comprising the step of determining whether each item in the metadata nesting table contains a group type.

22.   (Previously Presented) The method of claim 19 and further comprising the step of initializing a pattern-mapping table.

4

23.   (Previously Presented) The method of claim 22 and further comprising the step of directly mapping a link into the link table for each item in the metadata nesting table that does not contain a group type.

24.   (Previously Presented) The method of claim 23 and further comprising the step of creating an additional link table containing a mapping of a link pattern for each group type identified in the metadata item table.

25.   (Previously Presented) The method of claim 24 and further comprising the step of creating a create tuple loading action in the pattern mapping table associated with a particular pattern corresponding to each item in the item metadata table.

26.   (Previously Presented) The method of claim 25 wherein the loading step further comprises the step of creating an update tuple loading action in the pattern mapping table associated with a particular pattern corresponding to each attribute in the attribute metadata table.

27.   (Previously Presented) The method of claim 26 wherein the loading step further comprises the steps of:

creating a create tuple loading action in the pattern mapping table associated with a particular pattern corresponding to each group in a link; and

creating an assign action tuple loading action in the pattern mapping table associated with a particular pattern corresponding to each pair in the same link;

corresponding to each link in the link pattern table.

28.   (Previously Presented) The method of claim 27 wherein the loading step further comprises the step of forming a tree structure with the document data.

29.   (Previously Presented) The method of claim 28 wherein the loading step further comprises the step of traversing the formed tree and updating the at least one relational database table according to the rows of the pattern mapping table.

30.   (Previously Presented) The method of claim 1 and further comprising the step of optimizing the metadata.

5

31.   (Previously Presented) The method of claim 30 wherein the optimizing step further comprises the step of eliminating duplicate particle references in the metadata.

32.   (Previously Presented) The method of claim 31 wherein the optimizing step further comprises the step of simplifying references to corresponding elements, links and attributes in the metadata.

33.   (Previously Presented) The method of claim 32 wherein the optimizing step further comprises the step of inlining particular attributes of the metadata.

34.   (Previously Presented) The method of claim 1 wherein the document is an XML document.

35.   (Previously Presented) The method of claim 1 wherein the document-type definition is a DTD.

36.   (Previously Presented) The method of claim 1 wherein the data is tagged data.

37.   (Currently Amended) A system for generating a schema for a relational database corresponding to a document having a document-type definition and data complying with the document-type definition, the document-type definition having content particles representative of the structure of the document data, as well as loading the data into the relational database in a manner consistent with the relational schema, the system comprising:

an extractor adapted to read a document-type definition and to extract that extracts metadata representative of the document-type definition from the document-type definition, wherein the extractor stores the metadata in at least three tables comprising a metadata item table containing metadata representative of element types in the document-type definition, a metadata attribute table containing metadata representative of attributes in the document type definition, and a metadata nesting table containing metadata representative of nesting relationships between particles in the document type definition;

a generator operably interconnected to the extractor for automatically generating the schema for the relational database from the metadata, wherein at least one table is thereby

6

defined in the relational database corresponding to at least one content particle of the
document-type definition via the metadata; and

a loader operably interconnected to the generator for loading the document data into
the at least one table of the relational database according to the relational schema in a manner
driven by the metadata.

38.   (Cancelled)

39.   (Currently Amended) The system of claim 37 38 wherein the extractor creates
at least one default item in the item metadata table.

40.   (Previously Presented) The system of claim 39 wherein the extractor generates
a row in the item metadata table corresponding to each of the element type content particles
of the document-type definition.

41.   (Cancelled)

42.   (Currently Amended) The system of claim 40 41 wherein the extractor
generates a row in the attribute metadata table corresponding to each of the attribute type
content particles of the document-type definition.

43.   (Cancelled)

44.   (Currently Amended) The system of claim 42 43 wherein the extractor
generates a row in the nesting metadata table corresponding to each relationship identified in
the document-type definition between items identified in the item metadata table.

45.   (Previously Presented) The system of claim 44 wherein the generator creates
at least one table in the relational database schema of the relational database corresponding to
data in the metadata item table.

46.   (Previously Presented) The system of claim 45 wherein the generator alters the
schema of the relational database to add a columns to the at least one table of the relational
database schema corresponding to each row of the metadata attribute table.

7

47. (Previously Presented) The system of claim 46 wherein the generator alters the tables in the schema of the relational database to add columns representing links between tables of the relational database schema corresponding to each relationship identified in each row of the metadata nesting table.

48. (Previously Presented) The system of claim 47 wherein the generator alters the tables in the schema of the relational database by adding a foreign key to a parent table if a relationship identified between a pair of tables is a one-to-one relationship.

49. (Previously Presented) The system of claim 48 wherein the generator alters the tables in the schema of the relational database by adding a foreign key to a child table if a relationship identified between a pair of tables is a one-to-many relationship.

50. (Previously Presented) The system of claim 37 and further comprising a link table.

51. (Previously Presented) The system of claim 50 wherein the system determines whether each item in the metadata nesting table contains a group type content particle.

52. (Previously Presented) The system of claim 51 and further comprising a pattern-mapping table in an initialized state.

53. (Previously Presented) The system of claim 52 wherein the system directly forms a link in the link table for each item in the metadata nesting table that does not contain a group type.

54. (Previously Presented) The system of claim 53 wherein the loader creates an additional link table containing a mapping of a link pattern for each group type identified in the metadata item table.

55. (Previously Presented) The system of claim 54 wherein the system retrieves a preselected set of rows corresponding to each item in the metadata item table.

8

56.   (Previously Presented) The system of claim 55 wherein the system creates a create tuple loading action in the pattern mapping table associated with a particular pattern corresponding to each item in the item metadata table.

57.   (Previously Presented) The system of claim 56 wherein the system creates an update tuple loading action in the pattern mapping table associated with a particular pattern corresponding to each attribute in the attribute metadata table.

58.   (Previously Presented) The system of claim 57 wherein the system:

creates a create tuple loading action in the pattern mapping table associated with a particular pattern corresponding to each group in a link; and

creates an assign action tuple loading action in the pattern mapping table associated with a particular pattern corresponding to each pair in the same link;

wherein each created action corresponds to each link in the link pattern table.

59.   (Previously Presented) The system of claim 58 wherein the loader forms a tree structure with the document data.

60.   (Previously Presented) The system of claim 59 wherein the loader traverses the formed tree structure and updates the at least one relational database table according to the rows of the pattern mapping table.

61.   (Previously Presented) The system of claim 37 and further comprising an optimizer for refining the metadata.

62.   (Previously Presented) The system of claim 61 wherein the optimizer eliminates duplicate particle references in the metadata.

63.   (Previously Presented) The system of claim 62 wherein the optimizer simplifies references to corresponding elements, links and attributes in the metadata.

64.   (Previously Presented) The system of claim 37 wherein the document is an XML document.

65.   (Previously Presented) The system of claim 37 wherein the document-type definition is a DTD.

66.   (Previously Presented) The system of claim 37 wherein the data is tagged data.

67.   (Previously Presented) A system for generating a schema for a relational database corresponding to a document having a document-type definition and data complying with the document-type definition, the document-type definition having content particles representative of the structure of the document data, as well as loading the data into the relational database in a manner consistent with the relational schema, the system comprising:

an extractor adapted to read automatically a document-type definition that extracts metadata representative of the document-type definition from the document-type definition, wherein the extractor stores the metadata in at least three tables comprising a metadata item table containing metadata representative of element types in the document-type definition, a metadata attribute table containing metadata representative of attributes in the document type definition, and a metadata nesting table containing metadata representative of nesting relationships between particles in the document type definition; and

a generator operably interconnected to the extractor for automatically generating the schema for the relational database from the metadata, wherein at least one table is thereby defined in the relational database corresponding to at least one content particle of the document-type definition via the metadata.

68.   (Previously Presented) The system of claim 67 and further comprising a pattern-mapping table initially constructed in an initialized state.

69.   (Previously Presented) The system of claim 68 wherein the pattern mapping table is loaded with actions indicative of relationships between the data and the document-type definition.

70.   (Cancelled)

71.   (Previously Presented) The system of claim 67 wherein the generator forms a table with at least one default field in the relational database for each item contained in the metadata item table.

10

72. (Previously Presented) system of claim 71 wherein the generator forms a column in a corresponding table in the relational schema corresponding to each attribute in the metadata attribute table linked to an item in the metadata item table.

73. (Previously Presented) The system of claim 72 wherein the generator forms a link between tables in the relational database corresponding to nesting relationships contained in the metadata nesting table.

74. (Previously Presented) The system of claim 67 and further comprising a loader operably interconnected to the generator for loading the document data into the at least one table of the relational database according to the relational schema and driven by the metadata.

75. (New) The method of claim 33, wherein the step of inlining the particular attributes of the metadata reduces a number of nesting relationships and increases a number of attributes in the metadata.

76. (New) The method of claim 1, and further comprising optimizing the metadata by searching the metadata nesting table for a parent item and a child item having a one-to-one relationship where the child item does not have its own child, and inlining an attribute of the child item into an attribute of the parent item.

77. (New) The method of claim 76, wherein the optimizing step further includes removing the inlined attribute of the child item from the metadata item table.

11